

The email address you signed up with has not been verified. You won't be ranked on the leaderboard until you verify your account.

[Send Again](#)

Java If-Else ★

25 more points to get your first star!

Rank: **1956648** | Points: **0/25**



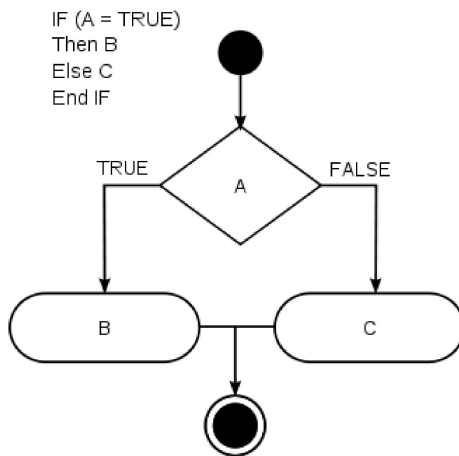
Problem

Submissions

Leaderboard

Editorial 

In this challenge, we test your knowledge of using if-else conditional statements to automate decision-making processes. An if-else statement has the following logical flow:



Source: [Wikipedia](#)

Task

Given an integer, ***n***, perform the following conditional actions:

- If ***n*** is odd, print **Weird**
- If ***n*** is even and in the inclusive range of **2** to **5**, print **Not Weird**
- If ***n*** is even and in the inclusive range of **6** to **20**, print **Weird**
- If ***n*** is even and greater than **20**, print **Not Weird**

Complete the stub code provided in your editor to print whether or not ***n*** is weird.

Input Format

A single line containing a positive integer, ***n***.

Constraints

- $1 \leq n \leq 100$

Output Format

Print **Weird** if the number is weird; otherwise, print **Not Weird**.

Sample Input 0

3

Sample Output 0

Weird

Sample Input 1

24

Sample Output 1

Not Weird

Explanation

Sample Case 0: $n = 3$

n is odd and odd numbers are weird, so we print `Weird`.

Sample Case 1: $n = 24$

$n > 20$ and n is even, so it isn't weird. Thus, we print `Not Weird`.

[Change Theme](#)

Language

Java 7



```
1  import java.io.*;
2  import java.math.*;
3  import java.security.*;
4  import java.text.*;
5  import java.util.*;
6  import java.util.concurrent.*;
7  import java.util.regex.*;
8
9  public class Solution {
10
11
12
13      private static final Scanner scanner = new Scanner(System.in);
14
15      public static void main(String[] args) {
16          int N = scanner.nextInt();
17          scanner.skip("(\\r\\n|[\\n\\r\\u2028\\u2029\\u0085])?");
18
19          scanner.close();
20      }
21  }
22
```

Line: 22 Col: 1

Upload Code as File

☐ Test against custom input

Run Code

Submit Code

The email address you signed up with has not been verified. You won't be ranked on the leaderboard until you verify your account.

[Send Again](#)

Basic Data Types ★

10 more points to get your first star!

Rank: **1279131** | Points: **0/10**



C++
C++

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial !\[\]\(b792654f2cef9719eabeb6c5be00811e_img.jpg\)](#)

Some C++ data types, their format specifiers, and their most common bit widths are as follows:

- Int ("%d"): 32 Bit integer
- Long ("%ld"): 64 bit integer
- Char ("%c"): Character type
- Float ("%f"): 32 bit real value
- Double ("%lf"): 64 bit real value

Reading

To read a data type, use the following syntax:

```
scanf("`format_specifier`", &val)
```

For example, to read a character followed by a double:

```
char ch;  
double d;  
scanf("%c %lf", &ch, &d);
```

For the moment, we can ignore the spacing between format specifiers.

Printing

To print a data type, use the following syntax:

```
printf("`format_specifier`", val)
```

For example, to print a character followed by a double:

```
char ch = 'd';  
double d = 234.432;  
printf("%c %lf", ch, d);
```

Note: You can also use cin and cout instead of scanf and printf; however, if you are taking a million numbers as input and printing a million lines, it is faster to use scanf and printf.

Input Format

Input consists of the following space-separated values: int, long, char, float, and double, respectively.

Output Format

Print each element on a new line in the same order it was received as input. Note that the floating point value should be correct up to 3 decimal places and the double to 9 decimal places.

Sample Input

```
3 12345678912345 a 334.23 14049.30493
```

Sample Output

```
3
12345678912345
a
334.230
14049.304930000
```

Explanation

Print int **3**,
followed by long **12345678912345**,
followed by char **a**,
followed by float **334.23**,
followed by double **14049.30493**.

[Change Theme](#)

Language

C++11



```
1  #include <iostream>
2  #include <cstdio>
3  using namespace std;
4
5  int main() {
6      // Complete the code.
7      return 0;
8  }
```

Line: 8 Col: 2

[Upload Code as File](#)

Test against custom input

[Run Code](#)[Submit Code](#)



Sum and Difference of Two Numbers ★

15 more points to get your first star!

Rank: 866349 | Points: 0/15



Problem

Submissions

Leaderboard

Editorial

Objective

The fundamental data types in c are int, float and char. Today, we're discussing int and float data types.

The `printf()` function prints the given statement to the console. The syntax is `printf("format string", argument_list);`. In the function, if we are using an integer, character, string or float as argument, then in the format string we have to write `%d` (integer), `%c` (character), `%s` (string), `%f` (float) respectively.

The `scanf()` function reads the input data from the console. The syntax is `scanf("format string", argument_list);`. For ex: The `scanf("%d",&number)` statement reads integer number from the console and stores the given value in variable ***number***.

To input two integers separated by a space on a single line, the command is `scanf("%d %d", &n, &m)`, where ***n*** and ***m*** are the two integers.

Task

Your task is to take two numbers of int data type, two numbers of float data type as input and output their sum:

1. Declare **4** variables: two of type int and two of type float.
2. Read **2** lines of input from stdin (according to the sequence given in the 'Input Format' section below) and initialize your **4** variables.
3. Use the **+** and **-** operator to perform the following operations:
 - Print the sum and difference of two int variable on a new line.
 - Print the sum and difference of two float variable rounded to one decimal place on a new line.

Input Format

The first line contains two integers.

The second line contains two floating point numbers.

Constraints

- $1 \leq \text{integer variables} \leq 10^4$
- $1 \leq \text{float variables} \leq 10^4$

Output Format

Print the sum and difference of both integers separated by a space on the first line, and the sum and difference of both float (scaled to **1** decimal place) separated by a space on the second line.

Sample Input

```
10 4
4.0 2.0
```

Sample Output

```
14 6
6.0 2.0
```

Explanation

When we sum the integers **10** and **4**, we get the integer **14**. When we subtract the second number **4** from the first number **10**, we get **6** as their difference. When we sum the floating-point numbers **4.0** and **2.0**, we get **6.0**. When we subtract the second number **2.0** from the first number **4.0**, we get **2.0** as their difference.

Change Theme Language: C



```
1  #include <stdio.h>
2  #include <string.h>
3  #include <math.h>
4  #include <stdlib.h>
5
6  int main()
7  {
8
9
10     return 0;
11 }
```

Line: 11 Col: 2

Upload Code as File

☐ Test against custom input

Run Code

Submit Code

[Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#)

The email address you signed up with has not been verified. You won't be ranked on the leaderboard until you verify your account.

[Send Again](#)

Dynamic Array ★

30 more points to get your first star!

Rank: **5174177** | Points: **0/30**



Problem Submissions Leaderboard Editorial 

- Declare a 2-dimensional array, **arr**, of **n** empty arrays. All arrays are zero indexed.
- Declare an integer, **lastAnswer**, and initialize it to **0**.
- There are **2** types of queries, given as an array of strings for you to parse:
 1. Query: 1 x y
 1. Let $idx = (x \oplus lastAnswer) \% n$.
 2. Append the integer **y** to **arr[idx]**.
 2. Query: 2 x y
 1. Let $idx = (x \oplus lastAnswer) \% n$.
 2. Assign the value **arr[idx][y % size(arr[idx])]** to **lastAnswer**.
 3. Store the new value of **lastAnswer** to an answers array.

Note: \oplus is the bitwise XOR operation, which corresponds to the ^ operator in most languages. Learn more about it on [Wikipedia](#). % is the modulo operator. Finally, size(arr[idx]) is the number of elements in arr[idx]

Function Description

Complete the dynamicArray function below.

dynamicArray has the following parameters:

- int n: the number of empty arrays to initialize in **arr**
- string queries[q]: query strings that contain 3 space-separated integers

Returns

- int[]: the results of each type 2 query in the order they are presented

Input Format

The first line contains two space-separated integers, **n**, the size of **arr** to create, and **q**, the number of queries, respectively.

Each of the **q** subsequent lines contains a query string, **queries[i]**.

Constraints

- $1 \leq n, q \leq 10^5$
- $0 \leq x, y \leq 10^9$
- It is guaranteed that query type **2** will never query an empty array or index.

Sample Input

```
2 5
1 0 5
1 1 7
1 0 3
2 1 0
2 1 1
```

Sample Output

7
3

Explanation

Initial Values:

$n = 2$

$lastAnswer = 0$

$arr[0] = []$

$arr[1] = []$

Query 0: Append **5** to $arr[(0 \oplus 0) \% 2] = arr[0]$.

$lastAnswer = 0$

$arr[0] = [5]$

$arr[1] = []$

Query 1: Append **7** to $arr[(1 \oplus 0) \% 2] = arr[1]$.

$arr[0] = [5]$

$arr[1] = [7]$

Query 2: Append **3** to $arr[(0 \oplus 0) \% 2] = arr[0]$.

$lastAnswer = 0$

$arr[0] = [5, 3]$

$arr[1] = [7]$

Query 3: Assign the value at index **0** of $arr[(1 \oplus 0) \% 2] = arr[1]$ to $lastAnswer$, print $lastAnswer$.

$lastAnswer = 7$

$arr[0] = [5, 3]$

$arr[1] = [7]$

7

Query 4: Assign the value at index **1** of $arr[(1 \oplus 7) \% 2] = arr[0]$ to $lastAnswer$, print $lastAnswer$.

$lastAnswer = 3$

$arr[0] = [5, 3]$

$arr[1] = [7]$

3

Change Theme

Language

C



```
1 #include <assert.h>
2 #include <ctype.h>
3 #include <limits.h>
4 #include <math.h>
5 #include <stdbool.h>
6 #include <stddef.h>
7 #include <stdint.h>
8 #include <stdio.h>
9 #include <stdlib.h>
10 #include <string.h>
11
12 char* readline();
```



```
13 char* ltrim(char*);
14 char* rtrim(char*);
15 char** split_string(char*);
16
17 int parse_int(char*);
18
19 /*
20  * Complete the 'dynamicArray' function below.
21  *
22  * The function is expected to return an INTEGER_ARRAY.
23  * The function accepts following parameters:
24  * 1. INTEGER n
25  * 2. 2D_INTEGER_ARRAY queries
26  */
```

Line: 220 Col: 1

[⬆ Upload Code as File](#)☐ [Test against custom input](#)[Run Code](#)[Submit Code](#)[Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#)

The email address you signed up with has not been verified. You won't be ranked on the leaderboard until you verify your account.

[Send Again](#)

Arithmetic Operators ★

35 more points to get your first star!

Rank: **2640642** | Points: **0/35**

[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)[Tutorial](#)

Check [Tutorial](#) tab to know how to solve.

Task

The provided code stub reads two integers from STDIN, ***a*** and ***b***. Add code to print three lines where:

1. The first line contains the sum of the two numbers.
2. The second line contains the difference of the two numbers (first - second).
3. The third line contains the product of the two numbers.

Example

***a* = 3**

***b* = 5**

Print the following:

```
8
-2
15
```

Input Format

The first line contains the first integer, ***a***.

The second line contains the second integer, ***b***.

Constraints

$$1 \leq a \leq 10^{10}$$

$$1 \leq b \leq 10^{10}$$

Output Format

Print the three lines as explained above.

Sample Input 0

```
3
2
```

Sample Output 0

```
5
1
6
```

Explanation 0

$$3 + 2 \Rightarrow 5$$

$$3 - 2 \Rightarrow 1$$

3 * 2 ⇒ 6

[Change Theme](#)

Language

Pypy 3



```
1  if __name__ == '__main__':  
2      a = int(input())  
3      b = int(input())
```

Line: 3 Col: 21

Upload Code as File**Test against custom input****Run Code****Submit Code**

[Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#)

The email address you signed up with has not been verified. You won't be ranked on the leaderboard until you verify your account.

Send Again

Select All ★

80 more points to get your first star!

Rank: 1755336 | Points: 0/80



Problem

Submissions

Leaderboard

Editorial

Query all columns (attributes) for every row in the **CITY** table.

The **CITY** table is described as follows:

CITY	
Field	Type
ID	NUMBER
NAME	VARCHAR2(17)
COUNTRYCODE	VARCHAR2(3)
DISTRICT	VARCHAR2(20)
POPULATION	NUMBER

DB2



```
1
2 ▾ /*
3     Enter your query here and follow these instructions:
4     1. Please append a semicolon ";" at the end of the query and enter your query in a single line to avoid
      error.
5     2. The AS keyword causes errors, so follow this convention: "Select t.Field From table1 t" instead of "select
      t.Field From table1 AS t"
6     3. Type your code immediately after comment. Don't leave any blank line.
7 */
```

Line: 1 Col: 1

[⬆️ Upload Code as File](#)

Run Code

Submit Code

The email address you signed up with has not been verified. You won't be ranked on the leaderboard until you verify your account. [Send Again](#)

Compare the Triplets ★

30 more points to get your first star!

Rank: **5174177** | Points: **0/30**



Problem Submissions Leaderboard Editorial 

Alice and Bob each created one problem for HackerRank. A reviewer rates the two challenges, awarding points on a scale from 1 to 100 for three categories: problem clarity, originality, and difficulty.

The rating for Alice's challenge is the triplet $a = (a[0], a[1], a[2])$, and the rating for Bob's challenge is the triplet $b = (b[0], b[1], b[2])$.

The task is to find their comparison points by comparing $a[0]$ with $b[0]$, $a[1]$ with $b[1]$, and $a[2]$ with $b[2]$.

- If $a[i] > b[i]$, then Alice is awarded 1 point.
- If $a[i] < b[i]$, then Bob is awarded 1 point.
- If $a[i] = b[i]$, then neither person receives a point.

Comparison points is the total points a person earned.

Given a and b , determine their respective comparison points.

Example

$a = [1, 2, 3]$

$b = [3, 2, 1]$

- For elements $a[0]$, Bob is awarded a point because $a[0] < b[0]$.
- For the equal elements $a[1]$ and $b[1]$, no points are earned.
- Finally, for elements $a[2]$, $a[2] > b[2]$ so Alice receives a point.

The return array is $[1, 1]$ with Alice's score first and Bob's second.

Function Description

Complete the function `compareTriplets` in the editor below.

`compareTriplets` has the following parameter(s):

- `int a[3]`: Alice's challenge rating
- `int b[3]`: Bob's challenge rating

Return

- `int[2]`: Alice's score is in the first position, and Bob's score is in the second.

Input Format

The first line contains 3 space-separated integers, $a[0]$, $a[1]$, and $a[2]$, the respective values in triplet a .

The second line contains 3 space-separated integers, $b[0]$, $b[1]$, and $b[2]$, the respective values in triplet b .

Constraints

- $1 \leq a[i] \leq 100$
- $1 \leq b[i] \leq 100$

Sample Input 0

```
5 6 7
3 6 10
```

Sample Output 0

```
1 1
```

Explanation 0

In this example:

- $a = (a[0], a[1], a[2]) = (5, 6, 7)$
- $b = (b[0], b[1], b[2]) = (3, 6, 10)$

Now, let's compare each individual score:

- $a[0] > b[0]$, so Alice receives **1** point.
- $a[1] = b[1]$, so nobody receives a point.
- $a[2] < b[2]$, so Bob receives **1** point.

Alice's comparison score is **1**, and Bob's comparison score is **1**. Thus, we return the array **[1, 1]**.

Sample Input 1

```
17 28 30
99 16 8
```

Sample Output 1

```
2 1
```

Explanation 1

Comparing the **0th** elements, **17 < 99** so Bob receives a point.

Comparing the **1st** and **2nd** elements, **28 > 16** and **30 > 8** so Alice receives two points.

The return array is **[2, 1]**.

[Change Theme](#)

Language

C



```

1  #include <assert.h>
2  #include <ctype.h>
3  #include <limits.h>
4  #include <math.h>
5  #include <stdbool.h>
6  #include <stddef.h>
7  #include <stdint.h>
8  #include <stdio.h>
9  #include <stdlib.h>
10 #include <string.h>
11
12 char* readline();
13 char* ltrim(char*);
14 char* rtrim(char*);
15 char** split_string(char*);
16
17 int parse_int(char*);
18
19 /*
20  * Complete the 'compareTriplets' function below.
21  *
22  * The function is expected to return an INTEGER_ARRAY.
23  * The function accepts following parameters:
24  * 1. INTEGER_ARRAY a
25  * 2. INTEGER_ARRAY b

```

26 * /

Line: 220 Col: 1

 Upload Code as File

☐ Test against custom input

Run Code

Submit Code

[Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#)